

MVIEW—A POWERFUL PRE- AND POST-PROCESSOR FOR TOUGH2

John Avis, Nicola Calder, and Robert Walsh

Geofirma Engineering

Ottawa, Ontario, K1R 1A2, Canada

javis@geofirma.com, ncalder@geofirma.com, rwalsh@geofirma.com

ABSTRACT

mView is a software tool used to pre- and post-process TOUGH2 models. Originally developed to assist in visualization of model results for the Yucca Mountain Project, mView has evolved into a powerful numeric modeling support system designed to process, analyze, and visualize complex geoscientific data. mView is model independent and supports a number of finite-difference (FD), finite-element (FE) and integral finite-difference (IFD) codes, including TOUGH2.

mView uses a toolkit paradigm for the modeling process. It does not presuppose a workflow, but provides the tools necessary for an analyst to address the specific requirements of their projects. These tools consist of a large number of objects that individually perform simple tasks, but can be linked in a data-flow network to function as complex algorithms. This approach provides extremely flexible capabilities for 2D and 3D gridding, property assignment, results analyses, and preparing complex 2D and 3D visualizations.

mView supports most common versions of TOUGH2 (all EOS) and TOUGH_MP. Rather than create a single monolithic input file, mView creates text files containing one or more TOUGH2 input blocks that can be simply combined with external batch programs. mView objects are available to create PARAM, ELEME, CONNE, ROCKS, GENER, INCON (up to 8 primary variables), FOFT, COFT, GOFT and TIME blocks. Time-varying boundary condition files (timbc.dat) for TOUGH_MP can also be created. For post-processing, mView reads time step (i.e. TIME block specified) gridblock and connection output and FOFT/COFT/GOFT files.

mView is implemented within an object-oriented application framework. Each mView object is an

independent testable entity and has been developed under an ISO9001:2008 software quality assurance program.

INTRODUCTION

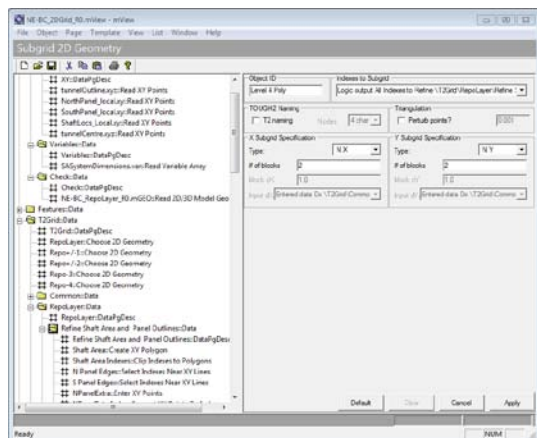
The TOUGH2 family of codes (Pruess et al., 1999) provides the modeler with extensive capabilities to simulate complex flow processes. However, it can be difficult to fully utilize the codes, since conventional groundwater flow and transport code pre- and post-processors are generally incompatible with TOUGH2 discretization and output. The few pre- and post-processors that are TOUGH2-specific are generally limited to using specific versions of the code in executable form only. Support is not available for all EOS, and user-customized versions of TOUGH2 may be difficult to incorporate into the workflow.

mView was originally created to provide visualization support for the Yucca Mountain Project, with the goal of integrating results from multiple models (TOUGH, FEHM, NUFT) in 3D visualizations (Avis et al. 1998, Avis et al. 2001). This original requirement led to the design choice that mView would be model agnostic—internal data structures were not tied to any particular model implementation (i.e., FD, FE, or IFV), but would rather support all implementations with generic data structures and file formats. This ensured that mView had integrated support for TOUGH2 specific data types. For example, most FE and FD codes provide flow output as vectors associated with elements (FE) or nodes (FD), while TOUGH2 defines flows over connections; all of these approaches are supported within mView.

mView has been built by modelers for use by modelers. At Geofirma, we have been using TOUGH2 for the last 10 years, primarily on radioactive-waste-related projects (Calder et al, 2006; Geofirma and Quintessa, 2011; Avis et al.,

SOFTWARE DESIGN

The mView user interface (UI) follows a folder-tree paradigm (Figure 1).



The user creates “pages,” which are analogous to folders. Objects are created and added to pages via a drop-down selection. The UI for each object is displayed in the main window adjacent to the tree when the object is first created or subsequently selected. The combina-

mView defines data types that can be communicated between objects. Example data types are: model geometries, model result data sets, scalars (arrays of values associated with a nodes or connections), and single real values.

Another mView innovation is the use of “data indexes.” These are data types that are essentially lists of geometry elements associated with gridblocks (nodes) or connections, which can be derived from model results (e.g., indexes of all nodes with $SL < 0.5$) or from pre-processing (e.g., all nodes of rock type X, all nodes within a polygon, all connections between specified node data indexes). Data indexes are used extensively in logical operations, property assignment, and in results presentation.

GRIDDING

There are four options for creating 2D grids: simple FD type regular rectangular, regular radial, irregular Voronoi, and irregular subgridded. Irregular grids are frequently used to minimize grid size by applying reduced discretization only where necessary. Voronoi grids respect the IFD formulation whereby connections are orthogonal to flow areas, but are difficult to create where precisely located property

transitions are required. Subgridding allows the combination and stepwise reduction of regular rectangular and radial grids and allows for easy implementation of local grid refinement that respects property boundaries but violates the orthogonality requirement. Our experience has shown that this leads to only very minor errors in calculated pressures. Figure 2 illustrates the difference between approaches in a portion of a relatively complex grid containing repository features.

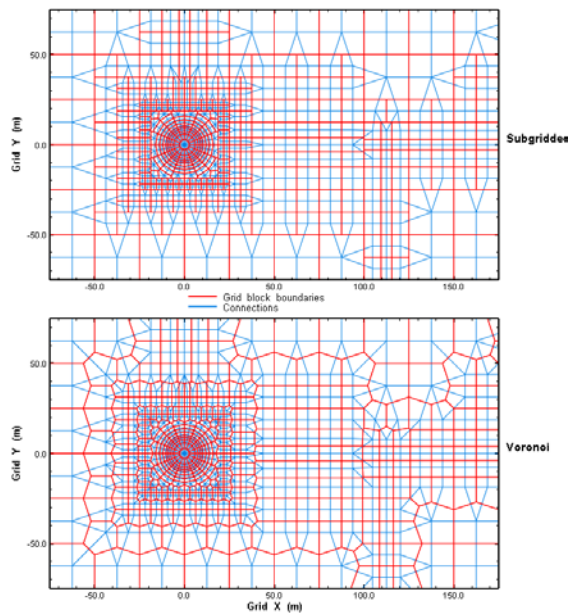


Figure 2. Comparison of subgridded and Voronoi 2D Grids.

3D grids are constructed as multiple layers of 2D grids. Layer orientation is usually in the horizontal (XY) plane, but can be vertical (XZ or YZ) if required. Creating a 3D grid requires definition of layer boundaries. mView can use existing digital elevation model (DEM) grids to specify layering coincident with geology. DEMs can also be created within mView using available kriging objects. Layers can also be defined by elevation or thickness. Each specified layer can be divided into multiple intermediate layers using fixed or adaptive spacing. Figure 3 is a cross section through an example 3D grid for a gas storage reservoir. Layers are specified at formation tops, with each formation subdivided into multiple intermediate layers.

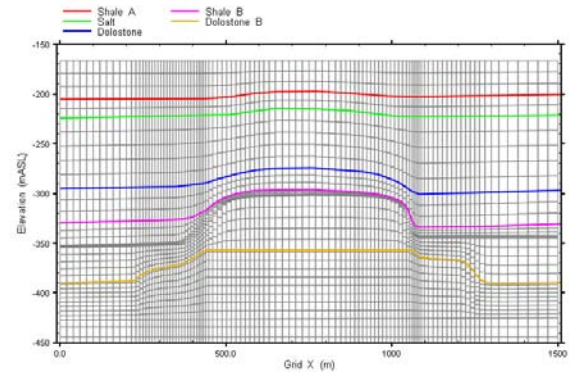


Figure 3. Slice through 3D grid showing layering from geologic surfaces.

3D grids can be combined to allow for variation in discretization by layer.

PROPERTY ASSIGNMENT

Property assignment is similarly flexible. A property set is created for a grid, and rock parameters entered in a spreadsheet format to define property groups. Groups are then assigned to model elements with a variety of objects. Basic objects include properties from DEM elevations, specified layers, constant elevations or within volumes and areas defined by polygons. Objects can be cascaded, for example to set basic layer properties from geology then to modify and add repository features or faults. Figure 4 shows a grid where formation data imported from a VULCAN geology model is used to set properties.

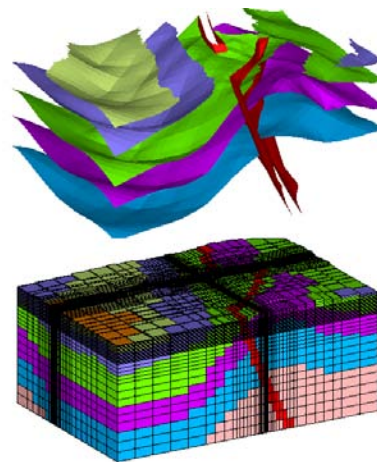


Figure 4. Geologic layers and property assignments.

Figure 5 illustrates property settings from polygons defining repository features. The figure also shows connections selected for COFT output specifications.

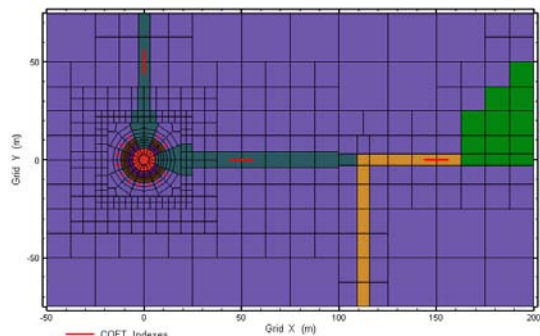


Figure 5. Repository feature property assignments from polygons and COFT connection indexes.

BOUNDARY AND INITIAL CONDITIONS

Fixed pressure/saturation boundary nodes are specified using data indexes. For example, nodes in top and bottom layers can be selected and combined. The data indexes can be used directly as negative volume (-VOLX) gridblocks in the TOUGH grid, or the block volume associated with those gridblocks can be specified to be an extremely large value. Initial conditions can be calculated from grid geometry (e.g., hydrostatic pressure or specified gradient) or specified by property (e.g., SG in repository nodes = 10.999). Results from other simulations, either on the same grid or a larger grid, can also be used. Figure 6 shows results from a previous steady-state flow simulation with repository operating period pressures imposed.

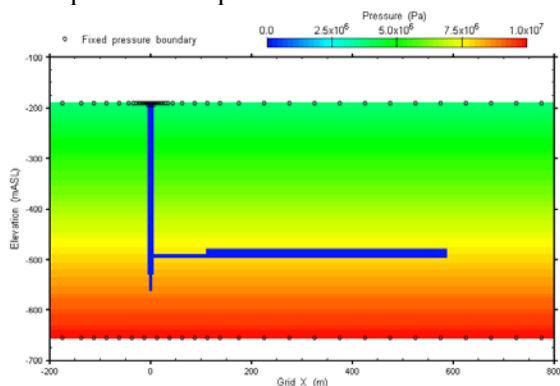


Figure 6. Initial pressure conditions and fixed pressure boundary nodes.

TOUGH2 INPUT FILE CREATION

As mentioned previously, mView objects are used to create text files comprising the major portions of a typical TOUGH2 input file. A brief review of these objects follows:

Create PARAM—provides a UI for most components of the PARAM block including MOP settings.

Create ELEME/CONNE/ROCKS—combines mView geometry and property data. ELEME and CONNE blocks are written to a single mesh file, while ROCKS are written separately. If IRP or ICP properties are defined, two-phase flow parameters are included in the ROCKS file. Default values for element volume and connection interface properties can be overridden. AHTX and PMX scalars can be incorporated. Automatic grid reorganization to incorporate negative volume boundary elements is performed if indicated.

Create INCON—Up to 8 primary variables may be specified. Optional PORX and INDICEE (for TMVOC) values can be used. Restart information from an existing SAVE file can be added.

Create FOFT/COFT/GOFT—creates file of gridblock identifiers from data index object input.

Create GENER—writes constant or time variable GENER for data indexes. Options allow rates to be allocated based on scaling factors. GX, EX and HX can be specified.

Create TIME—writes formatted table of output times from input table.

Create TIMBC—writes timvsp.dat file for TOUGH_MP time varying pressure boundary conditions.

mView-generated text files can be combined with other input files containing required TOUGH2 blocks (e.g., MULTI, START, DIFFU, ENDCY) using a simple batch command.

POST-PROCESSING

After a simulation is executed, mView is used to convert TOUGH2 text file output to an internal binary file format. Binary files require less disk space, can be accessed randomly to extract time step data, and are much faster to load. Versions of TOUGH2 used at Geofirma have been modified to include a binary file output option. File format and implementation details are available upon request.

After data conversion, post-processing objects are used to analyze and visualize model results.

There are over 200 objects that provide various analytic capabilities, from straightforward data extraction (select time step and variable as scalar, modify to include selected properties only), conversion (variable values at nodes/connections to time series tables, connection to node, connection to vector), math (symbolic arithmetic with scalars) and statistics (univariate on scalar, CDF calculation), to more complex processes (upscale results to different grids, stepwise regression). Additional objects are available to extract slices and layers from 3D grids for data analyses and plotting support, and to interpolate results from one grid to another. One of the major impediments to using mView is the plethora of available objects; the initial learning curve can be very steep. In the decade during which we have been using and developing mView, nearly every modeling project we have undertaken has resulted in the addition of one or more objects to perform specific tasks required to meet project goals.

Visual output is available as 2D or 3D plots. Plots can be further categorized as spatial (linear axes with fixed axes scale relationships, usually 1:1) and nonspatial (logarithmic or linear axes, no scale relationship).

The page and object UI paradigm extends to plots. When an mView plot page is created, a new top level window containing the plot appears on the user's desktop. Every object that is added to the plot page will cause a corresponding visual feature to appear on the plot. Plot objects can be categorized as data display (they appear within the axes limits and respect the axes scaling) or annotation objects (labels

and legends that can appear anywhere within the plot window). Data display plot objects include contouring, gridblock areas colored by node values or continuous shading over a nodal triangulation, vectors, isovolumes (3D), particle tracks and context information. Annotation plot objects include labels, grid lines, symbol/line legends, color bars, and data values.

Axes can be labeled and grid lines or ticks displayed. Alternatively, the appearance of the axes can be suppressed. Figure 7 shows the plot window for a 3D plot with multiple data display plot objects (solid colors for all nodes with $SG > 10^{-4}$, vertical and horizontal slices with pressures) and annotation objects (two color legends, a label showing selected time step) with no axes plotted.

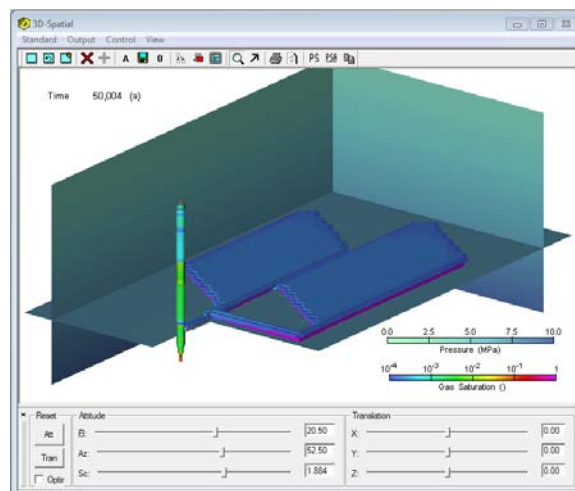


Figure 7. 3D plot window showing gas saturation and pressure.

Slider bars below the graphic window are used to precisely set the viewpoint and scale. The mouse wheel can also be used to rotate, tilt, and zoom. 3D plot objects can be visualized using transparency. This is useful for nested iso-volume plots.

A 2D plot window is shown in Figure 8. Data display plot objects have a “layer” setting which governs visibility. All objects on layer 0 are plotted first, followed by layer 1, layer 2, etc. In the figure, gas saturation is plotted on Layer 1, while permeability is plotted on Layer 0. Many data-display objects have a “report” capability that will indicate data identifiers and values under the cursor, also shown in the figure. This

capability is useful for diagnosing simulation problems and analyzing results.

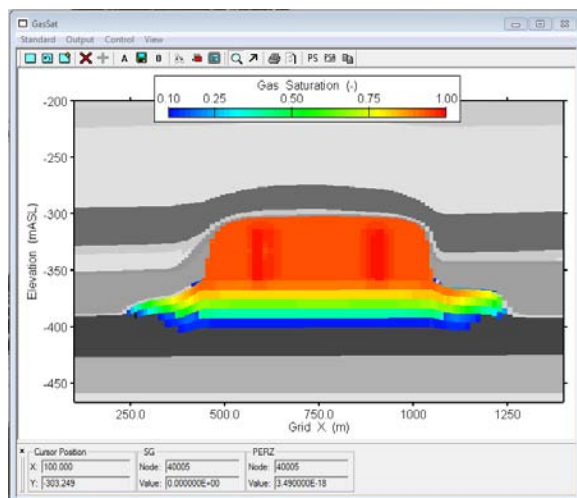


Figure 8. 2D plot window showing gas saturation and vertical permeability.

A final plot-page type is called a “Composite Plot.” These plots allow multiple 2D and 3D plot windows to be combined and embedded in a single window using flexible layout specifications, as shown in Figure 9.

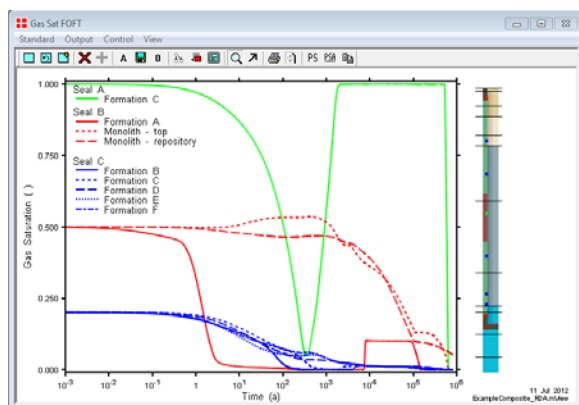


Figure 9. Composite plot with saturation versus time and monitoring point location plots

The graphics content of any plot window can be easily imported into reports using copy/paste or can be saved as jpeg or bitmap files. PostScript output files can be created when higher quality is required, such as on posters or large plots.

mView can be used to create animations. Plot windows can be set to write numerically sequenced bitmap files after each window update. These files can be processed by 3rd party animation software such as VideoMach

(www.gromada.com) to create Windows AVI files. Data objects have capabilities to step through sequences (e.g., time steps, grid layers) or to increment values (e.g., data interpolation times, iso-volume concentrations). 3D plot viewpoint parameters can also be varied over prescribed ranges to rotate or zoom in a 3D animation.

QUALITY ASSURANCE

mView was initially developed using C++ on Unix platforms under the Yucca Mountain Project DOE QA regime. Version 4.0 was qualified to these standards in 2004. Subsequently, the code was ported to Windows, which is the current development platform (Windows 7). mView is available in 32-bit and 64-bit versions for Windows XP or later. All code development has been performed under Geofirma Engineering's ISO 9001:2008 quality system, which contains procedures governing all software development activities. The procedures require unit testing and verification for all code. The mView architecture lends itself to testability since each object is independent of all other objects and can be easily verified. mView is currently undergoing qualification to the Canadian Nuclear Waste Management Organization (NWMO) software standards as Nuclear Grade software, used to support safety assessments.

CONCLUSIONS

mView is an powerful pre- and post-processor that provides a high level of support for TOUGH2 applications. It has been developed by modelers to support complex modeling projects and is continuously updated as needs evolve. It should be considered by the TOUGH community as an alternative to existing tools.

REFERENCES

- Avis, J.D., J.A. McNeish and S. Gurumurthy, *Scientific Visualization for Radioactive Waste Repository Analyses: mView*. EOS, Transactions American Geophysical Union Vol. 79, No. 45, pg.F24. 1998 Fall Meeting.
- Avis, J.D., Greenberg, H.R., and J.A. McNeish, *Visualization of Performance Assessment Process Model Results*. Presented at the American Nuclear Society 9th International High-Level Radioactive Waste Management

Conference (IHLRWM), Las Vegas, Nevada, April 2001.

Avis, J.D., P. Suckling, N. Calder, and R. Walsh, *T2GGM – a Coupled Gas Generation Model For Deep Geologic Disposal Of Radioactive Waste*, Proceedings TOUGH Symposium 2012, Lawrence Berkeley National Laboratory, Berkeley, California, 2012.

Calder, N., J.D. Avis, R. Senger and H. Leung, *Modifying TOUGH2 To Support Modelling Of Gas-Transport Through Saturated Compacted Bentonite As Part Of The Large-Scale Gas Injection Test (LASGIT) In Sweden*. Proceedings TOUGH2 Symposium, Berkeley, California, 2006.

Finsterle, S. *iTOUGH2 Command Reference, Version 4*. Report LBNL-40041, Earth Sciences Division, Lawrence Berkeley National Laboratory. University of California, Berkeley, USA. 1999

Geofirma and Quintessa, *Postclosure Safety Assessment: Gas Modelling*, Geofirma Engineering Ltd. and Quintessa Ltd. report for the Nuclear Waste Management Organization NWMO DGR-TR-2011-31, Toronto, Ontario, 2011.

Senger, R., P. Marschall, and S. Finsterle, *Investigation of Two-Phase Flow Phenomena Associated with Corrosion in an SF/HLW Repository in Opalinus Clay, Switzerland*, Physics and Chemistry of the Earth, vol. 33, p. S317-S326, 2007

Senger, R., J. Ewing, K. Zhang, J.D. Avis, P. Marschall, I. Gaus, *Modeling Approaches For Investigating Gas Migration From A Deep Low/Intermediate Level Waste Repository (Switzerland)*, Proceedings TOUGH Symposium 2009 Lawrence Berkeley National Laboratory, Berkeley, California, 2009

Pruess, K., C. Oldenburg, and G. Moridis, *TOUGH2 User's Guide, Version 2.0*, Report LBNL-43134, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1999.

Zhang, K., Wu, Y.S., Ding, C. & Pruess, K.: *TOUGH2_MP: A parallel version of TOUGH2*. Proceedings TOUGH Symposium 2003, Lawrence Berkeley National Laboratory, California, 2003.

GALLERY

